Creating Capsule Wardrobes from Fashion Images (Supplemental Material)

Wei-Lin Hsiao
UT-Austin
kimhsiao@cs.utexas.edu

Kristen Grauman
UT-Austin
grauman@cs.utexas.edu

This document consists of:

- Proof of Claim 3.2 in Section 3.1.2 of the main paper
- Implementation details for iterative-greedy algorithm.
- Computation complexity of naive and iterative greedy algorithm in Section 3.1.2 of the main paper
- Vocabulary for predicted attributes in Section 3.3 of the main paper
- Examples of false negative generated by randomly swapping pieces in Section 4.1 of the main paper
- Qualitative example images for most and least compatible outfits scored by baseline methods in Section 4.2 of the main paper
- Qualitative example images for personalized capsules obtained by nearest-neighbor baseline.
- Interface for human subject study in Section 4.2 of the main paper

1. Submodularity for objective function

Claim 3.2 *When fixing all other layers (i.e., upper, lower, outer) and selecting a subset of pieces one layer at a time, the probabilistic versatility coverage function in Eqn (3) is submodular, and the compatibility function in Eqn (2) is modular.*

Proof. Let A_j be candidate pieces from each layer j, where $j = \{0, 1, \ldots, m-1\}$, and D, B be any set such that $D \subseteq B$, $D = \prod_{j=0}^{m-1} A_j^D$, $B = \prod_{j=0}^{m-1} A_j^B$, where $A_j^D \subseteq A_j$, $A_j^B \subseteq A_j, \forall j$. Since $D \subseteq B$, $A_j^D \subseteq A_j^B, \forall j$.

Given a layer i, let $s_i \in A_i \setminus A_i^B$ be the piece additionally included. Outfits introduced by including s_i to B and D will be $O = \left\{s_i \times \prod_{j \neq i} A_j^B\right\}$ and $K = \left\{s_i \times \prod_{j \neq i} A_j^D\right\}$, respectively. Since $A_j^D \subseteq A_j^B$, $K \subseteq O$.

- Given a layer i, and fixing all other layers, versatility is submodular.

$$\begin{aligned} v_{B\cup O}(z_i) - v_B(z_i) &= 1 - \prod_{o_j \in B\cup O}(1 - P(z_i|o_j)) \\ &\quad - \left(1 - \prod_{o_j \in B}(1 - P(z_i|o_j))\right) \\ &= \prod_{o_j \in B}(1 - P(z_i|o_j)) - \prod_{o_j \in B\cup O}(1 - P(z_i|o_j)) \\ &= \prod_{o_j \in B}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in O}(1 - P(z_i|o_j))\right) \end{aligned}$$

Because $P(z_i|o_j)$ is defined as a probability, it is in the range $[0, 1]$, and therefore $(1 - P(z_i|o_j)) \in [0, 1], \forall j$. Since $D \subseteq B$, we have that $\prod_{o_j \in B}(1 - P(z_i|o_j)) \leq \prod_{o_j \in D}(1 - P(z_i|o_j))$. Thus,

$$\begin{aligned} &\prod_{o_j \in B}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in O}(1 - P(z_i|o_j))\right) \\ &\leq \prod_{o_j \in D}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in O}(1 - P(z_i|o_j))\right) \\ &= \prod_{o_j \in D}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in K}(1 - P(z_i|o_j)) \prod_{o_j \in O\setminus K}(1 - P(z_i|o_j))\right) \end{aligned}$$

When $O \setminus K = \varnothing$,

$$\begin{aligned} &v_{B\cup O}(z_i) - v_B(z_i) \\ &\leq \prod_{o_j \in D}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in K}(1 - P(z_i|o_j)) \prod_{o_j \in O\setminus K}(1 - P(z_i|o_j))\right) \\ &= \prod_{o_j \in D}(1 - P(z_i|o_j))\left(1 - \prod_{o_j \in K}(1 - P(z_i|o_j))\right) \\ &= \prod_{o_j \in D}(1 - P(z_i|o_j)) - \prod_{o_j \in D\cup K}(1 - P(z_i|o_j)) \\ &= v_{D\cup K}(z_i) - v_D(z_i) \end{aligned}$$

Since $K \subseteq O$, when $O \setminus K = \varnothing$, $O = K$, i.e. $\left\{s_i \times \prod_{j \neq i} A_j^B\right\} = \left\{s_i \times \prod_{j \neq i} A_j^D\right\}$, and thus $A_j^B = A_j^D, \forall j \neq i$. Submodularity is closed under

nonnegative linear combination, and user's personalized preference for each style i $w_i \geq 0$, thus $V(\mathbf{y})$ and $V'(\mathbf{y})$ are both submodular when given a layer i and fixing all other layers.

- Given a layer i, and fixing all other layers, compatibility is modular.

  Since $s_i \in A_i \setminus A_i^B$, $\left\{s_i \times \prod_{j\neq i} A_j^B\right\} \cap \left\{A_i^B \times \prod_{j\neq i} A_j^B\right\} = \varnothing$, i.e. $O \cap B = \varnothing$, and same with $D \cap K = \varnothing$.
  By $O \cap B = \varnothing$, we get

$$\begin{aligned} C(B \cup O) - C(B) &= \sum_{o_j \in B \cup O} c(o_j) - \sum_{o_j \in B} c(o_j) \\ &= \sum_{o_j \in B} c(o_j) + \sum_{o_j \in O} c(o_j) - \sum_{o_j \in B} c(o_j) \\ &= \sum_{o_j \in O} c(o_j) \end{aligned}$$

  and $C(D \cup K) - C(K) = \sum_{o_j \in K} c(o_j)$.

  By $K \subseteq O$, we have

$$\sum_{o_j \in O} c(o_j) = \sum_{o_j \in K} c(o_j) + \sum_{o_j \in O \setminus K} c(o_j)$$

  When $O \setminus K = \varnothing$,

$$\begin{aligned} C(B \cup O) - C(B) &= \sum_{o_j \in O} c(o_j) \\ &= \sum_{o_j \in K} c(o_j) + \sum_{o_j \in O \setminus K} c(o_j) \\ &= \sum_{o_j \in K} c(o_j) = C(D \cup K) - C(K) \end{aligned}$$

  Thus $C(\mathbf{y})$ is modular when given layer i and fixing all other layers.

■

2. Implementation details for iterative-greedy algorithm

We set our tolerance degree $\varepsilon = 0.5$, and find that our iterative-greedy algorithm typically converges after 5 iterations. We use Gibbs sampling [2] for compatibility inference. Due to the sampling process, $p(\boldsymbol{\theta}, \boldsymbol{z} | o_j, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta)$ fluctuates slightly at different run times. To increase robustness, we further apply a step function on our compatibility score $c(o_j)$, so that $c(o_j) \geq \epsilon$ is mapped to 1, and otherwise to 0. We fix $\epsilon = -4.69$, as validated in the compatibility experiment to give the best precision-recall trade-off.

3. Computation complexity for niave and greedy algorithms

Both the naive greedy and iterative greedy algorithms have a computation bottleneck when computing the objective $\mathbf{obj}(\mathbf{y_t})$. Its complexity is decided by N_i times $|\mathbf{y_t}|$, the size of the incrementally growing subset $\mathbf{y_t}$ at iteration t. In the following we show the algorithms of naive and iterative greedy, and analyze their $|\mathbf{y_t}|$ respectively. Without loss of generality, we assume our algorithms are provided with an initial piece for each layer i.

Naive greedy algorithm for submodular maximization, where $\mathbf{obj}(\mathbf{y}) := C(\mathbf{y}) + V(\mathbf{y})$.

for each time step $t = 1, 2, ...T$ **do**
  $\mathbf{y_{t-1}} = \prod_{i=0}^{m-1} A_{i(t-1)}$
  for each layer $i = 0, 1, ...m-1$ **do**
    $s_i^t := \text{argmax}_{s \in A_i \setminus A_{i(t-1)}} \delta_s$
    where $\delta_s = \mathbf{obj}(\mathbf{y_t}) - \mathbf{obj}(\mathbf{y_{t-1}})$
    where $\mathbf{y_t} = \mathbf{y_{t-1}} \cup \left\{s \times \prod_{j\neq i} A_{j(t-1)}\right\}$
  end for
end for

We need to compute $\mathbf{obj}(\mathbf{y_t})$ for all $s \in A_i \setminus A_{i(t-1)}$. At time step t and layer i, $|A_i \setminus A_{i(t-1)}| = N_i - (t+1)$. Since $N_i \gg (t+1)$ and is around the same scale for all i, we shorthand the term $N_i - (t+1)$ to N. Considering every candidate s and the set of outfits $\left\{s \times \prod_{j\neq i} A_{j(t-1)}\right\}$ introduced, computation for all sets introduced by all s becomes $N(t+1)^{(i-1)}$ times. Summing over all i and all t, the total computation is $\sum_{t=1}^{T} \sum_{i=0}^{(m-1)} N(t+1)^{(i-1)}$ times.

$$\begin{aligned} \sum_{t=1}^{T} \sum_{i=0}^{(m-1)} N(t+1)^{(i-1)} &= N \sum_{t=1}^{T} \frac{1-(t+1)^m}{1-(t+1)} \\ &= N \sum_{t=1}^{T} O(t^{m-1}) \end{aligned}$$

$\sum_{t=1}^{T} O(t^{m-1})$ is an $(m-1)$-th power series for the first T natural numbers. A closed formula at $m = 4$ equals $[\frac{T(T+1)}{2}]^2$, so the final complexity will be $O(NT^4)$ for naive greedy.

Algorithm 1 Proposed iterative greedy algorithm for submodular maximization, where $\mathbf{obj}(\mathbf{y}) := C(\mathbf{y}) + V(\mathbf{y})$.

$A_{iT} := \varnothing, \forall i$
$\Delta_{obj} := \varepsilon + 1$ ▷ ε is the tolerance degree for convergence
$\mathbf{obj}_{prev}^{m-1} := 0$
while $\Delta_{obj}^{m-1} \geq \varepsilon$ **do**
  for each layer $i = 0, 1, ...(m-1)$ **do**
    $A_{iT} = A_{i0} := \varnothing$ ▷ Reset selected pieces in layer i
    $\mathbf{obj}_{cur}^{i} := 0$
    for each time step $t = 1, 2, ...T$ **do**
      $\mathbf{y_{t-1}} = A_{i(t-1)} \times \prod_{i' \neq i} A_{i'T}$
      $s_i^{j_t} := \text{argmax}_{s \in A_i \setminus A_{i(t-1)}} \delta_s$ ▷ Max increment
      where $\delta_s = \mathbf{obj}(\mathbf{y_{t-1}} \uplus s) - \mathbf{obj}(\mathbf{y_{t-1}})$
      $A_{it} := s_i^{j_t} \cup A_{i(t-1)}$ ▷ Update layer i
      $\mathbf{obj}_{cur}^{i} := \mathbf{obj}_{cur}^{i} + \delta_{s_i^{j_t}}$
    end for
  end for
  $\Delta_{obj}^{m-1} := \mathbf{obj}_{cur}^{m-1} - \mathbf{obj}_{prev}^{m-1}$
  $\mathbf{obj}_{prev}^{m-1} := \mathbf{obj}_{cur}^{m-1}$
end while
procedure INCREMENTAL ADDITION ($\mathbf{y_t} := \mathbf{y_{t-1}} \uplus s$)
  $\mathbf{y_t^+} := s, s \in A_i \setminus A_{i(t-1)}$
  for $j \in \{1, \ldots, m\}, j \neq i$ **do**
    if $A_{jT} \neq \varnothing$ **then**
      $\mathbf{y_t^+} := \mathbf{y_t^+} \times A_{jT}$
    end if
  end for
  $\mathbf{y_t} := \mathbf{y_{t-1}} \cup \mathbf{y_t^+}$
end procedure

Let t_g denote at which iteration the while loop is. At iteration $t_g = 0$, for layer $i = 0$, each candidate piece s will introduce a set of outfits $\left\{ s \times \prod_{j \neq i} A_{jT} \right\}$. Since each layer has an initial piece, $|A_{jT}| = 1, \forall j$, and computation for the objective value of all sets introduced by all s is N times. After layer $i = 0$ selects T pieces, $|A_{0T}| = T$, and thus layer $i = 1$ computes TN times objective values. After that, layer $i = 2$ computes T^2N times, and so on. So at $g_t = 0$, the total computation complexity is $\sum_{i=0}^{m-1} T^i N$. For all iterations $t_g \geq 1$, we reset selected pieces at each layer i, and select T pieces again, so the complexity is $T^{(m-1)}N, \forall i$. Summing over all layers, we get $\sum_{i=0}^{m-1} T^{(m-1)} N = mNT^{(m-1)}$. At $m = 4$, we get computation complexity per t_g iteration $O(NT^3)$ for iterative greedy.

4. Vocabulary for catalog/outfit attributes

Tab. 1 lists the predicted attributes organized by types: *pattern*, *material*, *shape*, *collar*, *article*, *color*. We pair *pattern*, *material*, and *color* with body parts to get localized attributes. Since modeling correlation between attributes (e.g. *material translucent* co-occurs with *pattern lace*, *neckline scoop* co-occurs with *pattern graphics*) improves each individual attribute accuracy [5, 1], we subsample images from each type and multilabel them for catalog attribute prediction.

pattern	material	shape	collar	article	color
crochet	translucent	skirt drape pleated	scoop	T-shirt	white
camouflage	leather	skirt drape prairie	vneck	blouse	black
floral	denim	skirt drape flat	square	jacket	red
geo	fur	skirt length long	off-shoulder	blazer	pink
horizontal striped	down	skirt length medium	sweetheart	cardigan	orange
lace		skirt length short	turtle-neck	coat	yellow
leopard		skirt shape tight	shirt collar	vest	green
plaid		skirt shape loose		dress	blue
paisley		skirt shape full		skirt	purple
plain		pants loose		pants	brown
polka dot		pants flared		jeans	gray
tribal		pants peg-leg		leggings	beige
vertical striped		pants skinny		stocking	
zebra		pants short		boots	
		ruffle shirt		shoes	
		ruffle dress		sunglasses	
				hat	
				belt	
				scarf	
				bag	
				socks	
				sweater	

Table 1: Predicted attributes organized by types.

Figure 1: Left: outfits from exclusive meta-labels. Middle: randomly swapping pieces will form actually compatible outfits, i.e. those swapped within the same meta-label. Right: swapping pieces across exclusive meta-labels will be closer to true negatives.

5. Examples of false negatives

In Section 4.1 we describe our procedure to generate negative (not compatible) outfits for evaluation. Here we give more intuition about why this helps generate safe negatives. In Fig. 1 we show examples of outfits with different meta labels (*season*, *occasion*, *function*), and show negatives generated by swapping pieces from exclusive meta-labels, comparing with negatives randomly generated.

6. Qualitative example images for most/least compatible outfits predicted by baselines

Fig. 2 shows most and least compatible outfits predicted by baselines, Monomer [4], BiLSTM [3], along with ours (CTM). Most compatible outfits scored by us are those that consist of staples. Most compatibles scored by BiLSTM are those with a special pattern or material, which are more stylish. Most compatibles scored by Monomer contain mostly white pieces.

Figure 2: Most/least compatible outfits predicted by each method. Each row shows a method: our method tends to score outfits with staples as more compatible; BiLSTM [3] scores outfits more stylish as more compatible; Monomer [4] scores outfits with white pieces as more compatible.

7. Qualitative example images for personalized capsules obtained by nearest neighbor baseline

We predict attributes on users' outfits and all candidate pieces, and find the visually similar pieces (measured in attribute space) to those worn on each user. The result is shown in Fig. 3, comparing with the result using our method. Forming capsule wardrobes by using nearest neighbor does not take compatibility nor diversity into consideration, thus the results are mainly pieces similar in cut, shape, material and color.

8. Human subject interface

Fig. 4 and Fig. 5 show the interface of our human subject study on capsule wardrobes. In the instructions, we first describe the definitions of capsule wardrobes, and show examples of good and bad capsules, following explanations of why they are good and bad. In each question, we show (a) and (b) 2 candidate capsules, and ask subjects to choose which is better, where better is defined in the instructions: better capsules are those that can produce more compatible outfits. We ask subjects to avoid choosing EQUAL. Each question is also followed by confidence rating: from 1 = subtle to 3 = very obvious.

Figure 3: Personalized capsules tailored for user preference.

References

[1] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *ECCV*, 2012.

[2] J. Chen, J. Zhu, Z. Wang, X. Zheng, and B. Zhang. Scalable inference for logistic-normal topic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[3] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis. Learning fashion compatibility with bidirectional lstms. *ACM MM*, 2017.

[4] R. He, C. Packer, and J. McAuley. Learning compatibility across categories for heterogeneous item recommendation. In *ICDM*, 2016.

[5] K. Yamaguchi, T. Okatani, K. Sudo, K. Murasaki, and Y. Taniguchi. Mix and match: Joint model for clothing and attribute recognition. In *BMVC*, 2015.

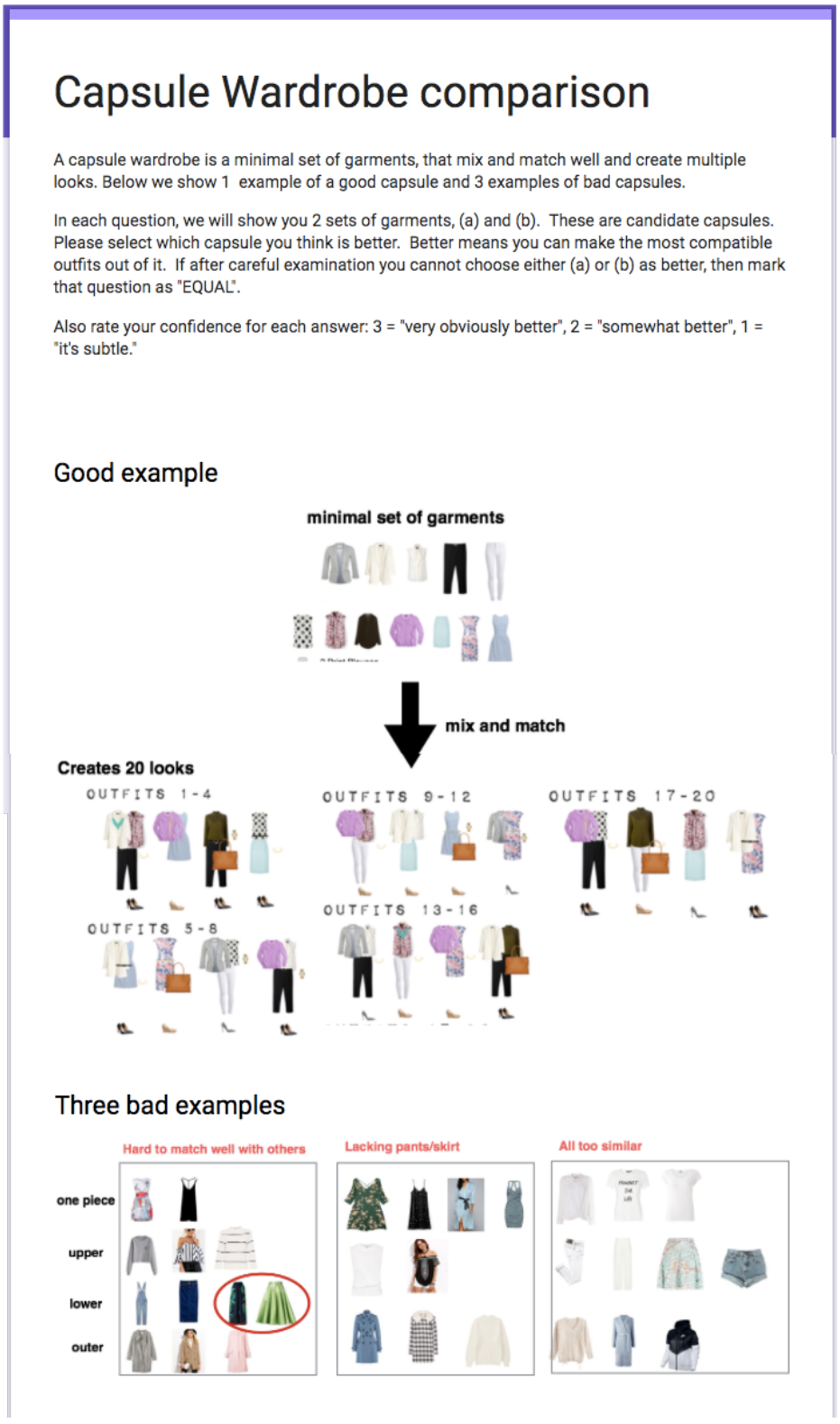

Figure 4: Instructions to guide human subjects: we show textual descriptions of capsule wardrobe definitions, and visual examples of good and bad capsules.

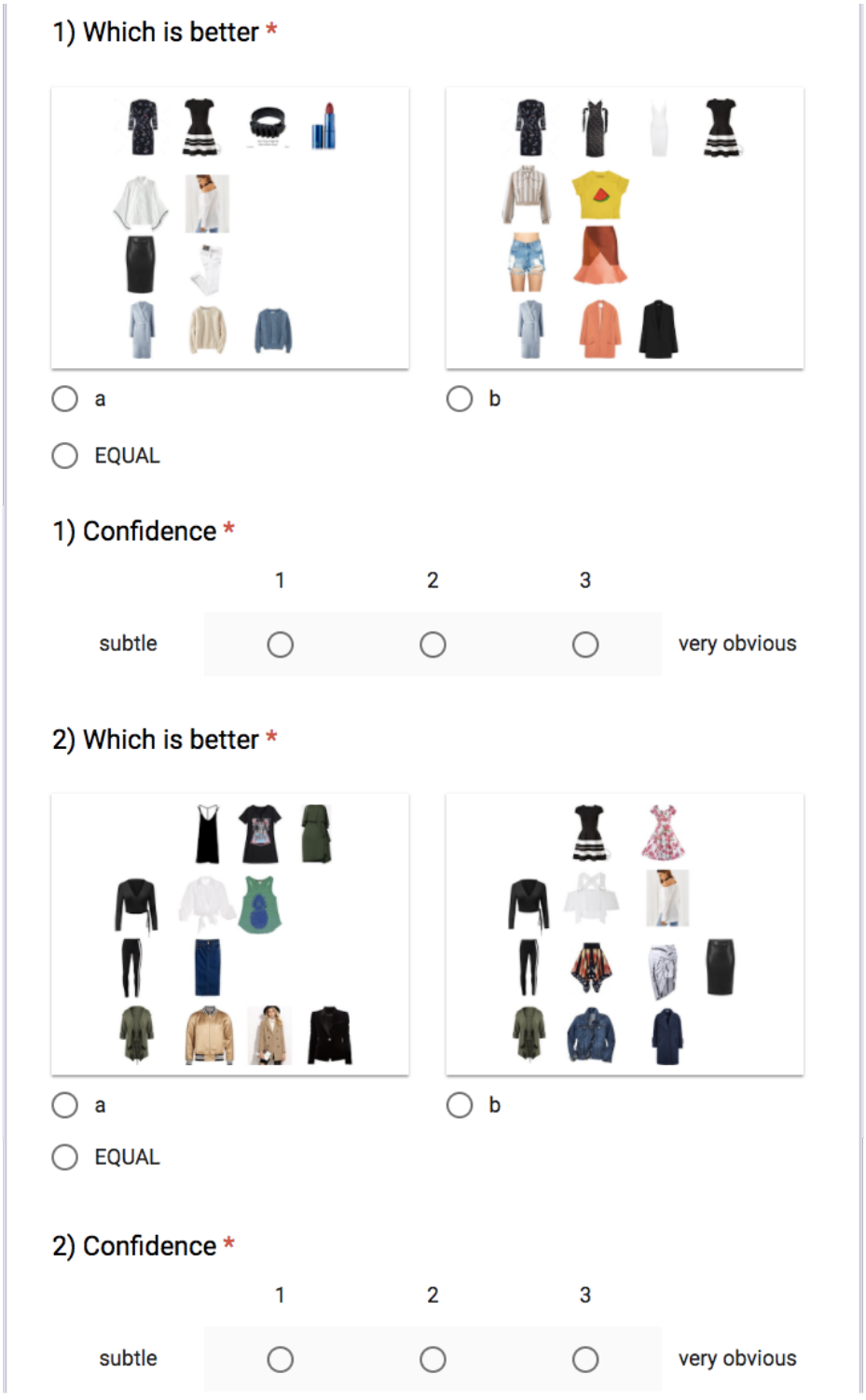

Figure 5: Questions shown to subjects: (a),(b) are sampled pairs of iterative vs. naive greedy capsules. We encourage subjects to avoid selecting EQUAL unless the difference between two capsules is too subtle to tell. Each comparison is followed by a confidence rating for provided answer. Best viewed on pdf.