Fully-Connected CRFs with Non-Parametric Pairwise Potentials

Supplementary Material

Neill D. F. Campbell
University College London

Kartic Subr
University College London

Jan Kautz
University College London

A. Approximate Euclidean Embedding

Given pairwise dissimilarities between $N = |\mathcal{P}|$ pixels in an image, and an integer $p \leq N$, we compute the p-dimensional feature space as an Euclidean embedding of the input dissimilarities. Let D be the input $N \times N$ dissimilarity matrix, such that

$$D_{ij} \quad = \quad \mathrm{d}\left(i, j, \mathcal{I}, \mathcal{T}\right), \tag{A.1}$$

and $\mathbf{F}$ be the $N \times p$ output configuration matrix. A p-dimensional Euclidean embedding of between N points in an unknown input space, is a canonical configuration of N points in $\mathcal{R}^p$ such that the Euclidean distance matrix of the points ideally equals D. Algorithms to discover such embeddings formulate the problem as optimizations and typically produce approximately Euclidean embeddings.

Classical Multidimensional Scaling (cMDS) cMDS [5] is an algebraic approach to solving the embedding problem. cMDS implicitly minimizes the Frobenius norm $\|K_{\mathrm{ref}} - K\|_F$, where $K = \mathbf{F}\mathbf{F}^T$ is the kernel (Gram) matrix of the reported (approximate) configuration, and K_{ref} is the Gram matrix of the ideal (exact) Euclidean embedding, when it exists.

For approximate, p-dimensional embeddings, $p < N$,

$$\mathbf{F}_{N\times p} \approx V_{N\times p}\sqrt{\Lambda_{p\times p}}, \tag{A.2}$$

where $\Lambda_{p\times p}$ has the p largest positive Eigenvalues of K along its diagonal and $V_{N\times p}$ contains the corresponding Eigenvectors. The presence of any negative Eigenvalues indicates that an exact Euclidean embedding does not exist. Classical MDS is equivelent to principal component analysis (PCA) when the input distances are Euclidean.

Landmark Multidimensional Scaling (LMDS) The $O(N^2)$ memory complexity of cMDS makes it an impractical choice when N is the number of pixels. Randomized approaches sample rows of the distance matrix to build approximate representatives of the entire matrix [4]. Depending on the coherence of the right singular vectors of D, it has been shown [2] that the Nyström approximation *is* the original matrix, under reasonable sampling conditions.

We use a Nyström approach called Landmark-multidimensional-scaling (LMDS) [1] which first embeds only $p + 1$ of the points (known as landmark points), for a p-dimensional embedding, using classical MDS. In practice, due to potential degeneracies, the number of landmark points needs to be $c > p + 1$ to ensure that they span the p-D space. The remaining points are triangulated from the embedded points using

$$\mathbf{f}_i = -\frac{1}{2}\, Y_{p\times c}\,(A_i - \bar{A}),\ c < i \leq N. \tag{A.3}$$

$Y_{p\times c}$ is the pseudo-inverse of $\mathbf{F}_{c\times p}$ and A_i is a c-vector containing distances of $\mathbf{f}_i$ from c landmark points. A_i is the i^{th} column of A (see Fig. A.1) and $\bar{A}$ is the average of the columns of A. Using (A.2), we can directly compute $Y_{p\times c} = V^T_{c\times p}\sqrt{\Lambda^{-1}_{c\times c}}$ where the p columns of $V_{c\times p}$ are Eigenvectors of K and $\sqrt{\Lambda^{-1}_{c\times c}}$ is a diagonal matrix containing the reciprocals of the square-roots of the Eigenvalues.

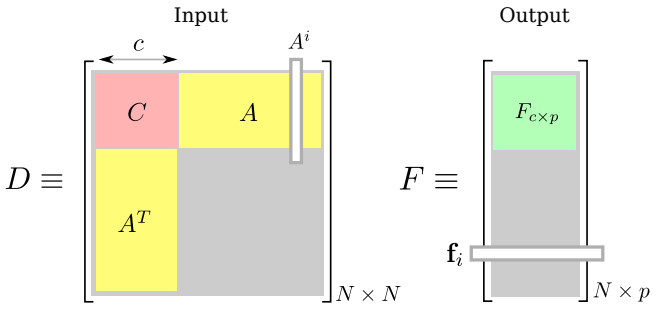
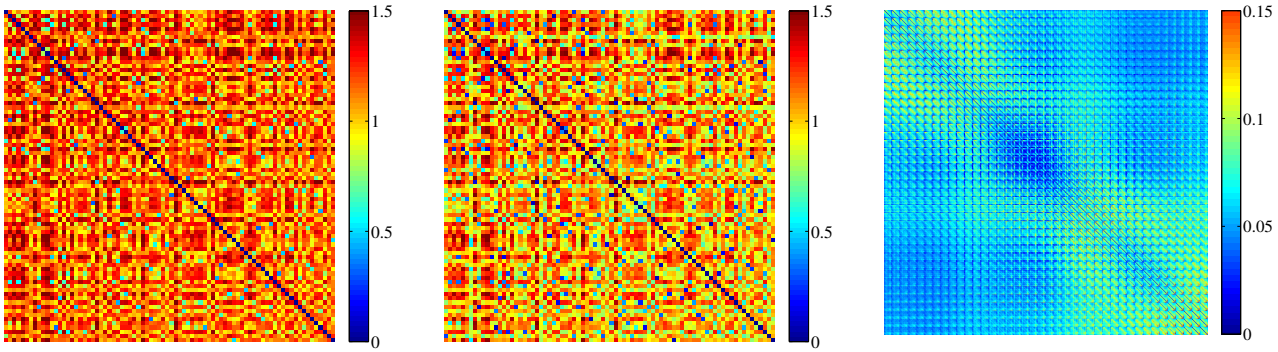


Figure A.1: We use LMDS [1] to compute feature vectors $\mathbf{f}_i$ from the stochastically sampled input pixel dissimilarity matrices C and A. First, the mutual dissimilarities between a few landmark points (block C) are embedded using classical MDS [5] to obtain $\mathbf{F}_{c\times p}$. The remaining $\mathbf{f}_i$ are then computed using (A.3).



(a) Target C matrix

(b) Estimated $\hat{C}$ matrix under the embedding

(c) Std. dev. of the resulting full dissimilarity matrix under random sampling

Figure B.1: Estimation of the feature space for an example in the KAIST Hanja2 database. (a) The target distances between randomly sampled landmarks in the test image. (b) The corresponding distances evaluated in the resulting embedded space. (c) The standard deviation of the complete distance matrix (for all N pixels) under embeddings generated by different random samples of the landmarks. Here $c = 80$ landmarks were used to generate an embedded space of dimension $p = 10$.

The time complexity of LMDS is $O(N(c + p^2) + c^3)$: $O(Nc)$ distance measurements in the first phase; $O(c^3)$ for the Eigendecomposition in the cMDS computation; and finally $O(Np^2)$ for embedding non-landmark points. The space complexity is $O(Nc)$. This is a key property to render the embedding feasible.

In summary, given the C and A matrices encoding the dissimilarity measures between a set of randomly chosen landmark points and the rest of the pixels in a test image, the embedding computes a set of feature vectors $\{\mathbf{f}_i\}$ whose mutual distances approximate the input dissimilarities. These feature vectors are used in the feature spaces for the Gaussian kernels that represent the pairwise potentials for each label. The embedding is agnostic of the source of the dissimilarities.

B. The KAIST Hanja2 Database Embedding

We also investigate the effect of changing the number of sampled landmarks c and limiting the dimensionality of the embedded space p. We note that it is a requirement that $c > p$; see § A. In Fig. 6(b) we find that increasing the number of landmarks provides a slight increase in performance whereas there is a dramatic decrease in performance if we limit the embedded space to a low number of dimensions. We note again that the number of dimensions may be found automatically by the embedding process, however, limiting the number of dimensions reduces the time requirements during inference and may therefore be advantageous. We also look at the estimated matrices directly in Fig. B.1.

The LMDS method is easily extensible to approximate Isomap [3] which is known to be a powerful non-linear dimensionality reduction technique. Given a sparse matrix D' that encodes each pixel's dissimilarities with its nearest neighbors, the C and A matrices that approximate Isomap are obtained by performing c single-source-shortest-path computations (from each of the c landmark points).

C. Further Details on Shape Completion Application Algorithm

In this section we provide a more formal definition of the algorithm we used to perform binary shape completion on the KAIST Hanja2 and Weizmann Horse datasets.

Objective

- Perform binary image completion in a masked region of a test image using a set of binary training images.

Input

- A set of R training images, $\mathcal{T} = \{\mathcal{T}_r\}_{r=1}^{R}$
- A test image $\mathcal{I}$ containing a masked region $\mathcal{M}$
- Query patch size q (for $q \times q$ pixel patches used to condition the pairwise potentials)
- Gaussian window std. deviation σ_w
- Trained dictionary patch size $Q > 3\,\sigma_w$ (for $Q \times Q$ pixel patches storing the non-parametric pairwise potentials)

Output

- The completed image $\hat{\mathcal{I}}$ with the masked region $\mathcal{M}$ filled with a binary output

Algorithm

- **Perform training for each class** $l \in \mathcal{L} =$ (foreground, background)
  - Extract all unique $q \times q$ patches from the training images $\mathcal{T}$ as $\{\mathbf{s}_b\}$ where

$$\mathbf{s}_b \quad := \quad \mathrm{Patch}(b, \mathcal{T}, q \times q) \tag{C.1}$$

    is the $q \times q$ patch, centered at b in training images $\mathcal{T}$, with a uniqueness constraint such that $\mathbf{s}_b \neq \mathbf{s}_{b'}\ \forall\ b' \neq b$.
  - Generate a training dictionary $\mathcal{D}^l(\mathbf{s}_b) \rightarrow \mathbf{S}_b^{(l)}$ where

$$\mathbf{S}_b^{(l)} \quad = \quad \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathrm{Window}\left[\mathrm{Patch}(v, \mathcal{T}^{(l)}, Q \times Q), \sigma_w\right] \tag{C.2}$$

    with

$$\mathcal{V} \quad = \quad \left\{ v \mid \mathrm{Patch}(v, \mathcal{T}^{(l)}, q \times q) = \mathbf{s}_b \right\} \quad , \tag{C.3}$$

    $\mathrm{Window}\,[\cdot, \sigma_w]$ as a Gaussian window (std. deviation σ_w), and $\mathcal{T}^{(l)}$ as the indicator image of class l in the training images $\mathcal{T}$. Thus $\mathbf{S}_b^{(l)}$ is an estimate for the density of class l in the neighborhood of $\mathbf{s}_b$.

- **Test with image $\mathcal{I}$**

  - For each class $l \in \mathcal{L} =$ (foreground, background)

    1. Take c samples $\Gamma = \{\gamma\}$ at random from the unmasked region $\gamma \in \{\mathcal{I}/\mathcal{M}\}$.
    2. Construct the $C^{(l)}$ and $A^{(l)}$ matrices from § 5 as

$$C_{ij}^{(l)} = D_{ij}^{(l)} \ , \ (i,j) \in \Gamma \tag{C.4}$$

$$A_{ij}^{(l)} = D_{ij}^{(l)} \ , \ i \in \Gamma \ , \ j \in \{\mathcal{I}/\Gamma\} \tag{C.5}$$

    using

$$D_{ij}^{(l)} = -\log\left[\exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}_j\|^2}{2\,\sigma_w^2}\right) P\left(x_j = l \mid x_i = l, \mathcal{I}, \mathcal{T}\right)\right] \tag{C.6}$$

$$= -\log\left[\exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}_j\|^2}{2\,\sigma_w^2}\right) \mathbf{S}_i^{(l)}(\mathbf{u}_j - \mathbf{u}_i)\right] \tag{C.7}$$

    as the dissimilarity between i and j, where $\mathbf{u}_i$ and $\mathbf{u}_j$ are the pixel coordinates, making use of the training dictionary to find $\mathbf{S}_i^{(l)} = \mathcal{D}^l(\mathbf{s}_i)$[1]. Here we use $\mathbf{S}_i^{(l)}(\mathbf{u}_j - \mathbf{u}_i)$ to denote a patch lookup.

    3. Use the LMDS algorithm (§ 5) to find $\mathbf{F}^{(l)}$, the matrix whose rows are the embedded vectors $\mathbf{f}_i^{(l)}$, from the matrices $C^{(l)}$ and $A^{(l)}$.

  - Run the mean-field inference algorithm (§ 3) using the feature spaces $\mathbf{F}^{(l)}$ in the model of

$$\mathrm{E}\left(\mathbf{x} \,|\, \mathcal{I}, \mathcal{T}\right) = \sum_{i \in \mathcal{P}} \psi(x_i) + \sum_{\substack{i,j \in \mathcal{P} \\ i \neq j}} \left[x_i \neq x_j\right] w \, \exp\left(-\frac{\|\mathbf{f}_i^{(x_i)} - \mathbf{f}_j^{(x_i)}\|^2}{2\,\sigma_f^2}\right) \ , \tag{C.8}$$

    with $\left[x_i \neq x_j\right]$ indicating a Potts model, to find the maximum posterior marginal solution for $\mathbf{x}$.

References

[1] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15.*, 2002. 1, 2

[2] A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nyström method. *Arxiv preprint arXiv:1004.2008*, 2010. 1

[3] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. 2

[4] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 13*, 2001. 1

[5] F. Young and R. Hamer. *Multidimensional scaling: History, theory, and applications*. L. Erlbaum Associates, 1987. 1, 2

[1] If the test patch $\mathbf{s}_i$ is not in the database of training patches $\{\mathbf{s}_b\}$ we use the nearest $\mathbf{s}_b$, under the Hamming distance function, to perform the lookup.