# Skeleton Clustering and Regression.

---

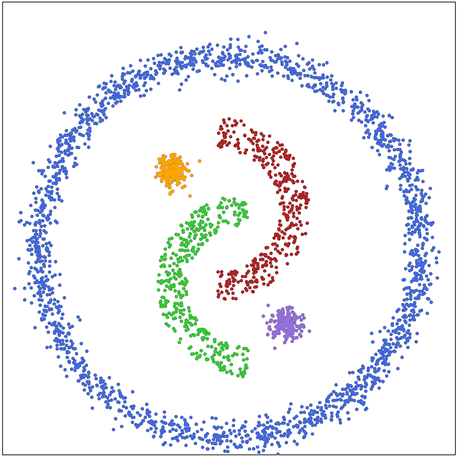
Yen-Chi Chen

Department of Statistics
University of Washington

◦ Supported by NSF DMS - 195278 and DMS - 2112907 and DMS - 2141808.

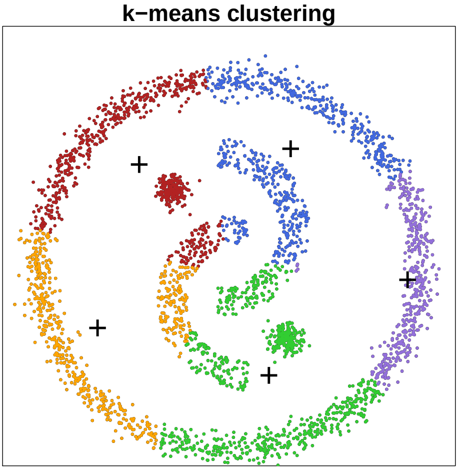
Joint work with Jerry (Zeyu) Wei

# A simple clustering problem in d=1000



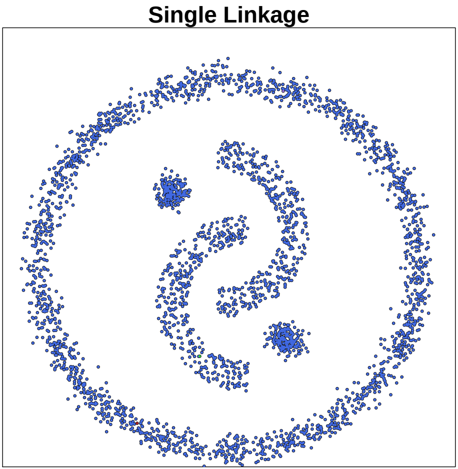
Data: $d = 1000$; only first 2 coordinates are shown here and the rest coordinates are Gaussian noises.

# A simple clustering problem in d=1000



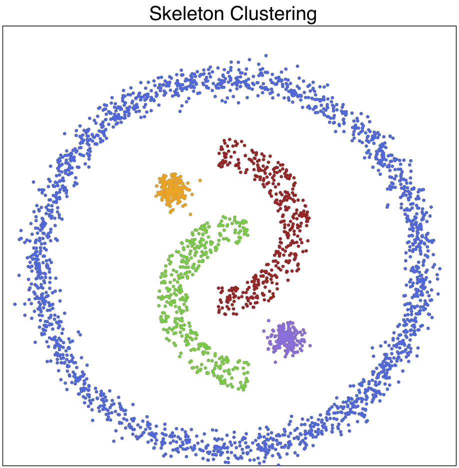

Data: $d = 1000$; only first 2 coordinates are shown here and the rest coordinates are Gaussian noises.

# A simple clustering problem in d=1000




Data: $d = 1000$; only first 2 coordinates are shown here and the rest coordinates are Gaussian noises.

# A simple clustering problem in d=1000

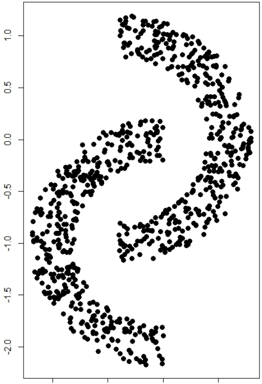



Data: $d = 1000$; only first 2 coordinates are shown here and the rest coordinates are Gaussian noises.

Idea:

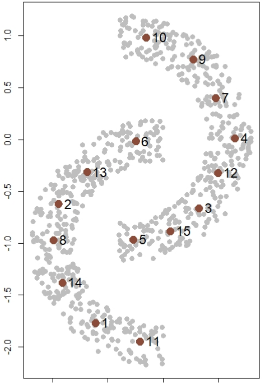
- We start with applying k-means clustering to the whole data with a large $k$.
- We then merge two clusters if they overlap a lot.

This procedure can be summarized as constructing a weighted graph called skeleton graph.



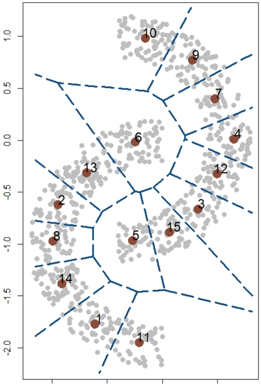

Original data.

# Skeleton Clustering - 2



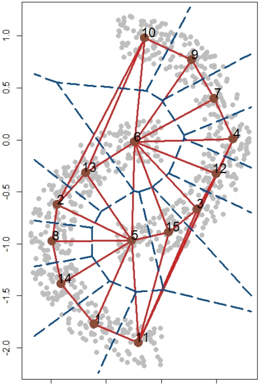

$k$-means: generating knots (centers of $k$-means clusters); $k = \sqrt{n}$.

# Skeleton Clustering - 2



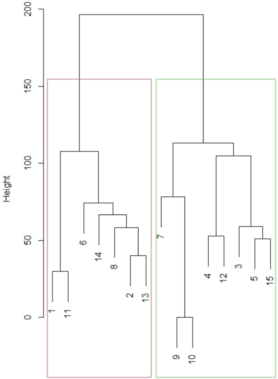

Voronoi cells of knots.

# Skeleton Clustering - 2



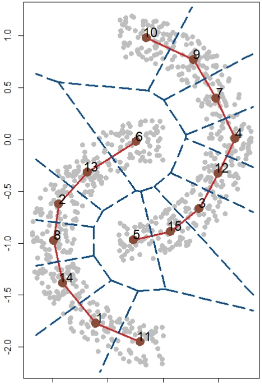

Delaunay triangulations–creating a graph. Assign a density-based weights on the edge to measure overlapping.

# Skeleton Clustering - 2



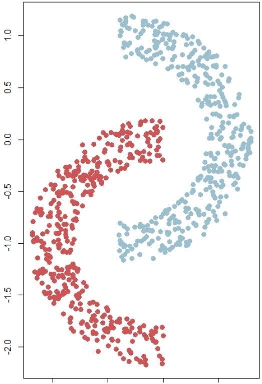

Convert a weighted graph into a dendrogram.

# Skeleton Clustering - 2




Cut the dendrogram to form the final clusters.

# Skeleton Clustering - 2




Final clustering result.

# Measuring the overlap between two knots

- Data: $X_1, \cdots, X_n \in \mathbb{R}^d$.
- Centers of k-means: $c_1, \cdots, c_k$.
- Partition of data:

$$\mathcal{X}_\ell = \{X_i : d(X_i, c_\ell) < d(X_i, c_j), \quad j \neq \ell\}$$

  for knot $c_\ell$.
- Goal: we want to create a quantity to measure the overlap between $c_j, c_\ell$.

# Voronoi density

- Intuition: $c_j$ and $c_\ell$ have a high overlap if
  1. they are close, i.e., $\|c_j - c_\ell\|$ is small,
  2. there are many observations between the $c_j$ and $c_\ell$
- Define the 2-NN region of $c_j$ and $c_\ell$:

$$A_{j\ell} = \{x : d(x, c_k) > \max\{d(x, c_j), d(x, c_\ell)\}, \forall k \neq j, \ell\}.$$

  Namely, the 2-NN of knots at $x$ is $c_j, c_\ell$.
- Let $\mathbb{P}_n(A) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A)$ be the empirical measure of the set $A$.
- We define the Voronoi density of $c_j, c_\ell$ as

$$\widehat{S}_{j\ell}^{VD} = \frac{\widehat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}.$$

# Voronoi density: remarks

- Recall that

$$A_{j\ell} = \{x : d(x, c_k) > \max\{d(x, c_j), d(x, c_\ell)\}, \forall k\},$$

$$\widehat{S}^{VD}_{j\ell} = \frac{\widehat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}.$$

- Clearly, $\widehat{S}^{VD}_{j\ell} = 0$ if $c_j$ and $c_\ell$ do not share a boundary.
- A population version of $\widehat{S}^{VD}_{j\ell}$ is

$$S^{VD}_{j\ell} = \frac{P(A_{j\ell})}{\|c_j - c_\ell\|},$$

where $P(A_{j\ell}) = P(X_i \in A_{j\ell})$. The convergence is fast assuming that knots are fixed.

# Alternatives: Face density

- The Voroni density is not the only way to measure the overlap.
- We also have some other good alternatives although the Voronoi density works the best in practice.
- Face density:

$$\widehat{S}^{FD}_{j\ell} = \widehat{\rho}_{j\ell}\left(\frac{1}{2}\right),$$

  where
  - $\widehat{\rho}_{j\ell}(t)$ is the KDE using observations in $\mathcal{X}_j, \mathcal{X}_\ell$ projected onto the line segment $\overline{c_j c_\ell}$ and evaluated at $t \cdot c_j + (1-t)c_\ell$.
  - Thus, $\widehat{\rho}_{j\ell}\left(\frac{1}{2}\right)$ is the midpoint (on the boundary).

# Alternatives: Tube density

- Similar to the face density, we also consider the tube density.
- Tube density:

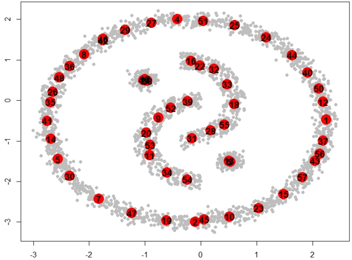
$$\widehat{S}^{TD}_{j\ell} = \min_{t\in[0,1]} \widehat{\omega}_{j\ell,R}(t),$$

  where
  - $\widehat{\omega}_{j\ell,R}(t)$ is the KDE of all observations projected to $\overline{c_j c_\ell}$ and evaluated at $t \cdot c_j + (1-t)c_\ell$ with a projected distance less than $R$.
  - Namely,

$$\widehat{\omega}_{j\ell,R}(t) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - t\cdot c_j - (1-t)c_\ell}{h}\right) I(d(X_i, \overline{c_j c_\ell}) < R).$$

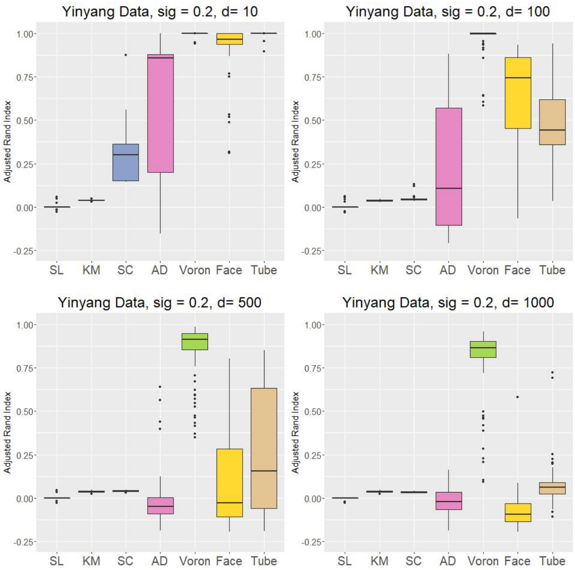
# Simulation: Yingyang

Knots for Yinyang Data, d=1000

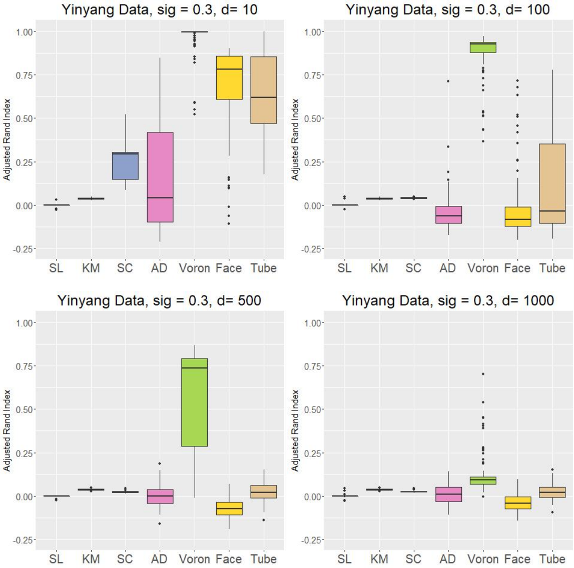


- An yingyang shape data with $n = 3200$.
- Main structure is 2D in the region $[-3, 3] \times [-3, 3]$.
- We added additional variables to make it high dimensions ($d = 10, 100, 500, 1000$).
- Additional noise level $\sigma = 0.2, 0.3$.
- We use adjusted rand index to evaluate performance.

# Simulation: Yingyang ($\sigma = 0.2$)



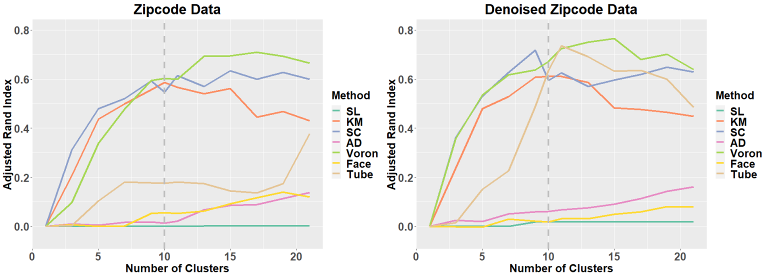
# Simulation: Yingyang ($\sigma = 0.3$)



# Real data: Zipcode data - 1

- $n = 2000$ with $d = 16 \times 16$ images of handwritten Hindu-Arabic numerals from.
- Numbers: 0,1,2,3,4,5,6,7,8,9.
- While this data is often used for classification, we remove the class label and treat it as a clustering problem.
- We consider using the original data and the denoised data (removing observations with the lowest 10% density).

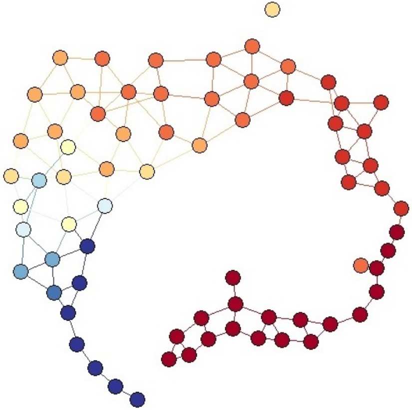
# Real data: Zipcode data - 2



# Skeleton Regression: Introduction

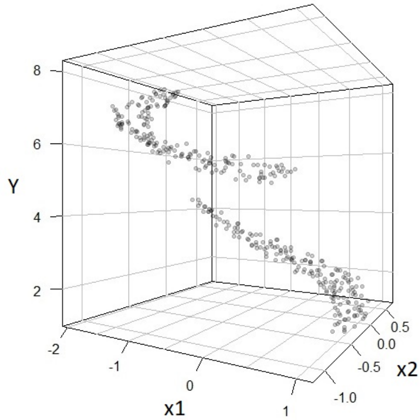
- The idea of skeleton can be used in a regression setting.
- Intuition: we construct skeleton using the covariates/features and do prediction on the skeleton.

# Skeleton Regression: Application to Astronomy data



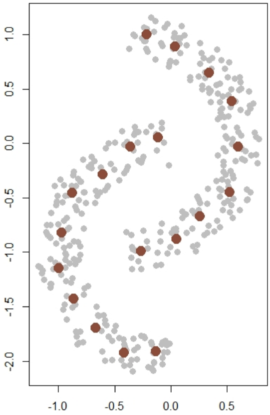
Detecting galaxy's redshift using color information (5D covariates).

# Skeleton Regression: big picture

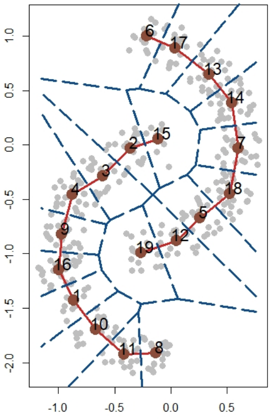



Raw data (2D covariate + 1D response).

# Skeleton Regression: big picture

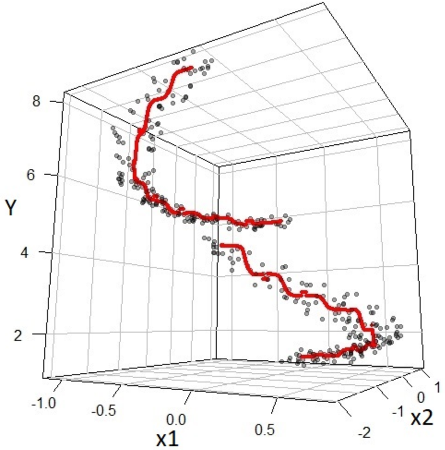



$k$-means: generating knots (centers of $k$-means clusters); $k = \sqrt{n}$.



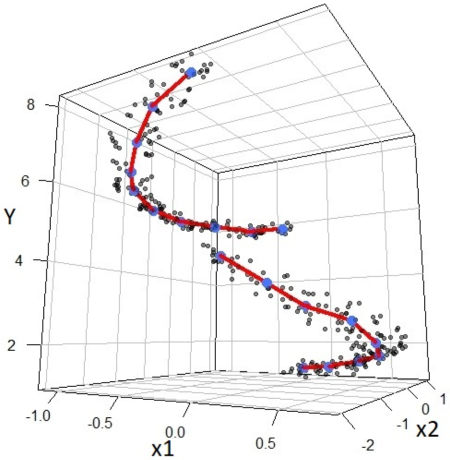

Generating the skeleton.

# Skeleton Regression: big picture




Skeleton kernel regression.

# Skeleton Regression: big picture




Skeleton linear spline.

# Skeleton Regression: Skeleton - 1

- Data: $(X_1, Y_1), \cdots, (X_n, Y_n) \in \mathbb{R}^d \times \mathbb{R}$.
- We construct skeletons using $X_1, \cdots, X_n$.
- Centers of k-means: $c_1, \cdots, c_k$.
- The skeleton clustering creates a weighted graph $G_0 = (V, W)$, where
  - $V = \{c_1, \cdots, c_k\}$
  - $W = \{w_{j\ell}\}$, where $w_{j\ell} = \widehat{S}^{VD}_{j\ell}$ is the Voronoi density.
- We choose a threshold $\lambda$ to convert it into an unweighted graph $G = (V, E)$, where $e_{j\ell} \in E$ ($j, \ell$ share an edge) if $W_{j\ell} \geq \lambda$.

# Skeleton Regression: Skeleton - 2

- The graph $G$ creates a skeleton $\mathcal{S} \subset \mathbb{R}^d$ such that
$$\mathcal{S} = V \cup \mathcal{E},$$
where
$$\mathcal{E} = \{tc_j + (1-t)c_\ell : t \in (0,1), e_{j\ell} \in E\}$$
denotes the edges.
- $\mathcal{S}$ is almost a 1D structure except for knots that may have multiple edges attaching to them.
- $\mathcal{S}$ can be decomposed into the vertex region $V$ and the edge region $\mathcal{E}$.
- We project each observation $X_i$ to $S_i \in \mathcal{S}$ and construct prediction models accordingly.

# Skeleton linear spline

- A simple nonparametric regression on skeleton is the linear spline.
- **Skeleton linear spline:** For each point $s \in \mathcal{S}$, we require the prediction model $m(s)$ that
  1. $m(s)$ is linear when $s$ is on an edge, and
  2. $m(s)$ is continuous at each knot.
- While it may looks non-trivial to fit this model, there is a simple representer theorem for this.

# Skeleton linear spline: representer theorem

- Define a regression model $m_\beta$ such that
  - $m_\beta(V_j) = \beta_j$ for each vertex,
  - $m_\beta(s) = t(s)\beta_j + (1 - t(s))\beta_\ell$ if $s = t(s)V_j + (1 - t(s))V_\ell$.
- Namely, the model is a linear interpolation of the prediction values on each knot.
- The model $m_\beta$ is determined by the coefficients $\beta_1, \cdots, \beta_k$ on the knots.

**Theorem (Wei and Chen (2023))**

*Any skeleton linear spline model can be written as $m_\beta$ for some $\beta$.*

# Skeleton linear spline: fitting

- Fitting the skeleton linear spline is very easy.
- For every observation $X_i$ with a projected location $S_i \in \mathcal{S}$, we further convert it into a vector $Z_i \in [0,1]^k$ such that

$$Z_{ik} = \begin{cases} 1 & \text{if } S_i = V_k \text{ is on the vertex} \\ t & \text{if } S_i = tV_k + (1-t)V_\ell \text{ for some } V_\ell \\ 0 & \text{otherwise.} \end{cases}$$

- With this, the prediction value $m_\beta(S_i) = Z_i^T \beta$.
- Thus, when we estimate $\beta$ using the least square, this becomes a linear regression problem with an analytic solution:

$$\widehat{\beta} = (ZZ^T)^{-1} ZY.$$

# Metric space induced by skeleton

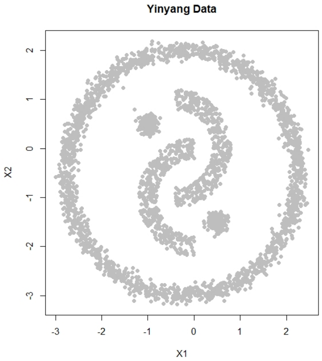
- The set $\mathcal{S}$ is equipped with a metric $d_{\mathcal{S}}$ because
  - each vertex $c_j \in \mathbb{R}^d$ has a location in Euclidean space and
  - each edge $e_{j\ell}$ has a length $\|c_j - c_\ell\|$.
- For two points $s_1, s_2 \in \mathcal{S}$, their distance $d_{\mathcal{S}}(s_1, s_2)$ will be the shortest distance in $\mathcal{S}$. If they belong to two different connected component, we set $d_{\mathcal{S}}(s_1, s_2) = \infty$.
- The metric space $(\mathcal{S}, d_{\mathcal{S}})$ allows us to use a wide variety of methods for prediction.

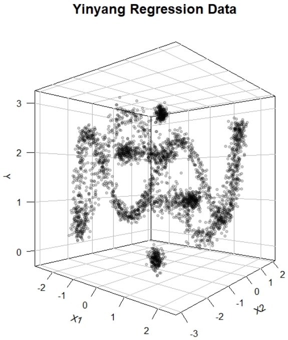
# Skeleton kernel regression

- As a classical example, we may use kernel regression on the skeleton.
- The prediction value $\widehat{m}_h(s)$ is

$$\widehat{m}_h(s) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{d_{\mathcal{S}}(s,S_i)}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{d_{\mathcal{S}}(s,S_j)}{h}\right)}.$$

- Other methods such as kNN is applicable as well.




We add additional covariates to make it a high-dimensional data.

# Skeleton Regression: simulations - 2

| Method | Medium SSE (5%, 95%) | nknots | Parameter |
| --- | --- | --- | --- |
| kNN | 204.5 (192.3, 221.9) | - | neighbor=18 |
| Ridge | 2127.0 (2100.2, 2155.2) | | $\lambda = 7.94$ |
| Lasso | 1556.8 (1515.4, 1607.9) | | $\lambda = 0.0126$ |
| SpecSeries | 1506.4 (1469.1,1555.6) | - | bandwidth = 2 |
| S-Kernel | 112.8 (102.0, 121.7) | 38 | bandwidth = 6 $r_{hns}$ |
| S-kNN | 139.6 (129.6,148.7) | 38 | neighbor = 36 |
| S-Lspline | 95.8 (88.6, 102.6) | 38 | - |

$d = 1000$. We use 10-fold cross-validation for every method.

# Conclusion

- Skeleton approach offers a flexible framework.
- It shows promising results in both clustering and regression when the number of covariates is high.
- However, a couple of open questions remains:
  - Understanding the effect of $k$-means when $k$ is large.
  - How does the randomness of $k$-means affects the final result.
  - Principled way to post-process the knots.
- Main references:
  - Skeleton clustering: arXiv 2104.10770
  - Skeleton regression: arXiv 2303.11786

# Thank You!

More details can be found in
http://faculty.washington.edu/yenchic.