Tracking

Dr. Xiqun Lu
College of Computer Science
Zhejiang University

Minimum Output Sum of Squared Error Filter (MOSSE) [1]

- The MOSSE filter is training **online**.
- MOSSE finds a filter ***h*** that minimizes the sum of squared error between the actual output of the convolution and the desired output of the convolution. The minimization problem takes the form:

$$\boxed{\min_{\mathbf{H}^*} \sum_i \left| \mathbf{F}_i \odot \mathbf{H}^* - \mathbf{G}_i \right|^2}$$

where $\odot$ denotes the Hadamard product.

MOSSE [1]

- Set the partial derivative of the above error function w.r.t. **H** equals to zero, we have

$$E = \sum_i \left| \mathbf{F}_i \odot \mathbf{H} - \mathbf{G}_i \right|^2 = \sum_i (\mathbf{F}_i \odot \mathbf{H} - \mathbf{G}_i)^H (\mathbf{F}_i \odot \mathbf{H} - \mathbf{G}_i)$$

$$\frac{\partial E}{\partial \mathbf{H}} = \sum_i \mathbf{F}_i^H (\mathbf{F}_i \odot \mathbf{H} - \mathbf{G}_i) = 0$$

$$\mathbf{H}^* = \frac{\sum_i \mathbf{F}_i^H \odot \mathbf{G}_i}{\sum_i \mathbf{F}_i^H \odot \mathbf{F}_i}$$

- Regularization

$$\mathbf{H}^* = \frac{\sum_i \mathbf{F}_i^H \odot \mathbf{G}_i}{\sum_i \mathbf{F}_i^H \odot \mathbf{F}_i + \varepsilon}$$

where ε is the regularization parameter. This result suggests that adding the energy spectrum of **the background noise** to that of the training imagery will produce a filter with better in noise tolerance.

Updating —— Running Average

$$\boxed{\begin{aligned}\mathbf{H}_i^* &= \frac{\mathbf{A}_i}{\mathbf{B}_i}\\ \mathbf{A}_i &= \eta \mathbf{G}_i \odot \mathbf{F}_i^H + (1-\eta)\mathbf{A}_{i-1}\\ \mathbf{B}_i &= \eta \mathbf{F}_i \odot \mathbf{F}_i^H + (1-\eta)\mathbf{B}_{i-1}\end{aligned}}$$

where η is the learning rate. This puts more weight on recent frames and lets the effect of previous frames decay exponentially over time.

Kernelized Correlation Filter (**KCF**) [2]

- Ridge regression
  - It admits a simple closed-form solution
  - Can achieve performance that is close to SVM
  - The goal of training is to find a function $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z}$ that minimizes the squared error over sample $\mathbf{x}_i$ and their regression targets y_i,

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2$$

$$= \sum_{i=1}^{N} (\mathbf{w}^T\mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|^2$$

$$= \sum_{i=1}^{N} (\mathbf{x}_i^T\mathbf{w} - y_i)^2 + \lambda \|\mathbf{w}\|^2$$

$$= \left\| \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} \mathbf{w} - \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \right\|^2 + \lambda \|\mathbf{w}\|^2$$

$$Let \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix}_{N\times M} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}_{N\times 1}$$

$$= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2$$

$$= (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$$

$$\frac{\partial E}{\partial \mathbf{w}} = \mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w} = 0 \rightarrow \mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y} \qquad (2)$$

N — the number of training samples

If each training sample $\mathbf{x}$ has the dimension of M, the computational complexity of this ridge regression is $O(M^3)$, since the main computational load is to compute $(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}$.

Kernelized Correlation Filter (**KCF**) [2]

- Cyclic shifts
- Permutation matrix

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} \qquad P\mathbf{x} \to \begin{pmatrix} x_N \\ x_1 \\ \vdots \\ x_{N-2} \\ x_{N-1} \end{pmatrix} \qquad (4)$$

  - Each column has one and only one “1”, each row has one and only one “1”
  - We can chain *u* shifts to achieve a larger translation by using the matrix power $P^u\mathbf{x}$.

Kernelized Correlation Filter (**KCF**) [2]

- Circularly shifted samples

(a) (b) (c) (d) (e) (f)

(a) The base sample **x**, (b) – (f) circularly shifted versions

- Due to the cyclic property, we get the same signal **x** periodically every M shift.

$$\left\{ P^u \mathbf{x} \mid u = 0, 1, \cdots, M-1 \right\} \qquad (5)$$

- Cyclic shifts will induce **distortion** to samples, except the base sample **x**, the other circularly shifted samples are not the true negative samples but the virtual samples.
  - However, this undesirable property can be mitigated by appropriate **padding** and **windowing.**

Circulant Matrix

- To compute a regression with shifted samples, we can use the set of Eq.(5) as the rows of a data matrix **X**:

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_m \\ x_m & x_1 & x_2 & \cdots & x_{m-1} \\ x_{m-1} & x_m & x_1 & \cdots & x_{m-2} \\ \vdots & & & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{pmatrix}$$

- Since the circulant matrix can be diagonalized by the DFT

$$\mathbf{X} = F diag(\hat{\mathbf{x}}) F^H \quad (7)$$

where $\hat{\mathbf{x}}$ denotes the DFT of the base signal $\mathbf{x}$, $\hat{\mathbf{x}} = \mathrm{F}(\mathbf{x})$

Ridge Regression

- The DFT matrix F is a unitary matrix, and unitary matrix preserves the 2-norm.

$$E = \left\| \mathbf{X}\mathbf{w} - \mathbf{y} \right\|^2 + \lambda \left\| \mathbf{w} \right\|^2$$

$$= \left\| F\mathbf{X}\mathbf{w} - F\mathbf{y} \right\|^2 + \lambda \left\| F\mathbf{w} \right\|^2$$

$$= \left\| F\mathbf{X}F^H F\mathbf{w} - F\mathbf{y} \right\|^2 + \lambda \left\| F\mathbf{w} \right\|^2 \qquad (F^H F = I)$$

$$= \left\| \hat{\mathbf{X}}\hat{\mathbf{w}} - \hat{\mathbf{y}} \right\|^2 + \lambda \left\| \hat{\mathbf{w}} \right\|^2 = (\hat{\mathbf{X}}\hat{\mathbf{w}} - \hat{\mathbf{y}})^H (\hat{\mathbf{X}}\hat{\mathbf{w}} - \hat{\mathbf{y}}) + \lambda \hat{\mathbf{w}}^H \hat{\mathbf{w}}$$

$$\frac{\partial E}{\partial \hat{\mathbf{w}}} = -\hat{\mathbf{X}}^H (\hat{\mathbf{X}}\hat{\mathbf{w}} - \hat{\mathbf{y}}) + \lambda \hat{\mathbf{w}} = 0$$

$$\hat{\mathbf{X}} = diag(\hat{\mathbf{x}}) \qquad \hat{\mathbf{X}}^H = diag(\hat{\mathbf{x}}^*)$$

$$\hat{\mathbf{w}} = (\hat{\mathbf{X}}^H \hat{\mathbf{X}} + \lambda I)^{-1} \hat{\mathbf{X}}^H \hat{\mathbf{y}} = (diag(\hat{\mathbf{x}}^*) diag(\hat{\mathbf{x}}) + \lambda I)^{-1} (diag(\hat{\mathbf{x}}^*) \hat{\mathbf{y}})$$

$$= diag\left(\frac{\hat{\mathbf{x}}^*}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \right) \hat{\mathbf{y}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}$$

The computational complexity is O(Mlog$M$).

Nonlinear Regression

- Kernel trick — Mapping the inputs of a linear problem to a non-linear feature space φ(**x**) with the kernel trick consists of:
  - 1) Expressing the solution **w** as a linear combination of the samples:

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$$

    The variables under optimization are thus **α**, instead of **w**.

  - 2) The dot-products are computed using kernel function κ (e.g. Gaussian and Polynomial)

$$\varphi^T(\mathbf{x})\varphi(\mathbf{x}') = \kappa(\mathbf{x}, \mathbf{x}')$$

    The dot-products between all pairs of samples are usually stored in a $N \times N$ **kernel matrix** K, with elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

    The regression function's complexity grows with the number of samples,

$$f(\mathbf{z}) = \mathbf{w}^T \mathbf{z} = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{z}, \mathbf{x}_i) \qquad (15)$$

Fast Kernel Regression

- The kernelized version of ridge regression is given by

$$\boldsymbol{\alpha} = (K + \lambda I)^{-1}\mathbf{y} \qquad (16)$$

- In general, the kernel matrix K is not circular.
- **Theorem 1**. Given circulant data matrix $C(\mathbf{x})$, the corresponding kernel matrix K is circulant if the kernel function satisfies $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(P\mathbf{x}, P\mathbf{x}')$, for any permutation matrix P.
  - Radial Basis Function kernels — e.g., Gaussian
  - Dot-product kernels — e.g., linear, polynomial

Fast Kernel Regression

- Knowing which kernels we can use to make *K* circulant, it is possible to diagonalize Eq.(16) as in the linear case:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda} \qquad (17)$$

$\mathbf{k}^{\mathbf{xx}}$ is the first row of the kernel matrix $K = C(\mathbf{k}^{\mathbf{xx}})$.

$$\mathbf{k}_i^{\mathbf{xx}} = \varphi^T(\mathbf{x})\varphi(P^{i-1}\mathbf{x})$$

$\hat{\mathbf{k}}^{\mathbf{xx}}$ is the kernel correlation of x with itself, in the Fourier domain.

Fast Detection

- To detect the target, we typically wish to evaluate $f(\mathbf{z})$ on several locations around the estimated location in the previous frame, i.e., for several candidate patches. These patches can be modeled by cyclic shifts.
- Denote by $K^{\mathbf{z}}$ the (asymmetric) kernel matrix between all training samples and all candidate patches. Since the samples and patches are cyclic shifts of base sample $\mathbf{x}$ and base patch $\mathbf{z}$, respectively, each element of $K^{\mathbf{z}}$ is given by $\kappa(P^{i-1}\mathbf{z}, P^{i-1}\mathbf{x})$.
- It is easy to verify that this kernel matrix satisfies Theorem 1, and is circulant for appropriate kernels. $K^{\mathbf{z}} = C(\mathbf{k}^{\mathbf{xz}})$ where $\mathbf{k}^{\mathbf{xz}}$ is the kernel correlation of $\mathbf{x}$ and $\mathbf{z}$.

$$\hat{f}(\mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}} \qquad (22)$$

References

- [1] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui, “Visual object tracking using adaptive correlation filters,” in Proc. of CVPR, 2010.
- [2] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filter” IEEE Trans. On Pattern Analysis and Machine Intelligence, vol.37, no.3, pp.583-596, Mar. 2015.

Thank You

Dr. Xiqun Lu
xqlu@zju.edu.cn